Discovering Frequent Episodes and Learning Hidden Markov Models: A Formal Connection

Srivatsan Laxman, P.S. Sastry, *Senior Member*, *IEEE*, and K.P. Unnikrishnan

Abstract—This paper establishes a formal connection between two common, but previously unconnected methods for analyzing data streams: discovering frequent episodes in a computer science framework and learning generative models in a statistics framework. We introduce a special class of discrete Hidden Markov Models (HMMs), called Episode Generating HMMs (EGHs), and associate each episode with a unique EGH. We prove that, given any two episodes, the EGH that is more likely to generate a given data sequence is the one associated with the more frequent episode. To be able to establish such a relationship, we define a new measure of frequency of an episode, based on what we call nonoverlapping occurrences of the episode in the data. An efficient algorithm is proposed for counting the frequencies for a set of episodes. Through extensive simulations, we show that our algorithm is both effective and more efficient than current methods for frequent episode discovery. We also show how the association between frequent episodes and EGHs can be exploited to assess the significance of frequent episodes discovered and illustrate empirically how this idea may be used to improve the efficiency of the frequent episode discovery.

Index Terms—Temporal data mining, sequential data, frequent episodes, Hidden Markov Models, statistical significance.

1 Introduction

Data sets with temporal dependencies frequently occur in business, engineering, and scientific scenarios. Over the years, many data mining techniques for analyzing such data streams have been proposed [1], [2], [3], [4], [5], [6]. The general techniques for such analysis can be broadly classified into two approaches: pattern discovery and learning generative models.

Searching for interesting or frequently occurring patterns has attracted a lot of attention in temporal data mining [1], [2], [5]. The central idea in frequent pattern discovery is to seek expressive patterns and fast discovery algorithms that render the technique both useful as well as efficient in the data mining context. The patterns sought could be, e.g., temporally ordered sequences of attribute values [1], [2], [6] or such sequences with more sophisticated structure [7], [8], [9].

Learning generative models is another important perspective in time-series analysis. Hidden Markov Models (HMMs) constitute a rich class of models that are popularly used for describing time-series data. Many such Markovian models have been used in a variety of applications [3], [10], [11], [12], [13], [14].

Overall, techniques for pattern discovery are often more useful for data summarization and rule generation applications. Most such techniques use counting-type arguments and have what may be called a computer science viewpoint. Model-based techniques, on the other hand, use stochastic methods and have a statistical framework. These techniques provide a principled approach to describing/modeling the statistics that govern data generation. For data mining, both approaches are important and should be used to complement each other [15]. This paper is motivated by such considerations.

This paper establishes a formal connection between a pattern discovery framework based on frequent episodes and a class of generative models based on HMMs. We define a specialized class of HMMs and correspond each episode with a unique HMM in this class. We prove that, given any two episodes, the HMM associated with the more frequent episode is more likely to generate the data and vice versa. This allows one to rigorously relate frequent episodes with HMMs. To our knowledge, this is the first instance of such a formal connection and it has interesting consequences. For example, we show that it gives us a mechanism to test whether a frequent episode discovered is, in a sense, statistically significant. This also leads to a mechanism whereby a reasonable frequency threshold can be automatically calculated, thus leading to what may be termed parameterless data mining.

A second contribution of this paper is a new frequency measure for episodes, namely, the number of nonoverlapping occurrences of an episode, and an algorithm for obtaining the corresponding set of frequent episodes. It is this new frequency count which makes it possible to formally connect frequent episode discovery and HMM learning. In addition, it also significantly speeds up the frequent episode discovery process. We illustrate all these through some simulations.

The rest of the paper is organized as follows. Section 2 provides a brief overview of frequent episodes in event streams. Section 3 presents our new frequency measure and the counting algorithm. Section 4 proves our main results connecting episodes and HMMs. In Section 5, we discuss the

- *S. Laxman and P.S. Sastry are with the Indian Institute of Science, Bangalore 560012 India. E-mail: {srivats, sastry}@ee.iisc.ernet.in.*
- *K.P. Unnikrishnan is with Manufacturing Systems Research, General Motors R & D Center, 30500 Mound Road, Warren, MI 48090-9055. E-mail: k.unnikrishnan@gm.com.*

Manuscript received 14 July 2004; revised 24 Dec. 2004; accepted 5 Apr. 2005; published online 19 Sept. 2005.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0212-0704.

consequences of this formal connection. Simulation results are presented in Section 6 and conclusions in Section 7.

2 Frequent Episodes in Event Streams

This section briefly introduces the framework of frequent episode discovery [1]. The data is a sequence of events, $\langle (E_1, t_1), (E_2, t_2), \ldots \rangle$, where E_i represents an *event type* and t_i the *time of occurrence* of the ith event. The E_i take values from a finite set of event types. For example, the following is an event sequence containing eight events:

$$\begin{aligned} &\langle (A,1), (B,3), (D,4), (C,6), \\ &(E,12), (A,14), (B,15), (C,17) \rangle. \end{aligned} \tag{1}$$

An *episode* is an ordered tuple of event types. (In the formalism of [1], this corresponds to the *serial episode*). For example, $(A \rightarrow B \rightarrow C)$ is a 3-node episode. An episode is said to *occur* in an event sequence if there are events in the sequence with the same time ordering as specified by the episode. In the example sequence (1), the events $\{(A,1), (B,3), (C,6)\}$ constitute an occurrence of the episode $(A \rightarrow B \rightarrow C)$, while $\{(A,14), (B,3), (C,6)\}$ do not.

A *subepisode* is a subsequence of the episode which has the same ordering as the episode. For example, $(A \rightarrow B)$, $(A \rightarrow C)$, and $(B \rightarrow C)$ are 2-node subepisodes of the 3-node episode $(A \rightarrow B \rightarrow C)$, while $(B \rightarrow A)$ is not.

The *frequency* of an episode can be defined in many ways. A reasonable frequency count must guarantee that any subepisode is at least as frequent as the episode.

A frequent episode is one whose frequency exceeds a user-specified threshold. The procedure for discovering frequent episodes proposed in [1] is based on the same general idea as the Apriori algorithm [2]. This method exploits the fact that a necessary condition for an N-node episode to be frequent is that all its $(N-1)$-node subepisodes should also be frequent. We use the same general procedure and, since such a procedure for candidate generation is fairly standard in data mining algorithms, we omit the details here.

Calculating the frequencies of a set of candidate episodes is the main computationally intensive step in frequent episode discovery. In [1], the frequency of an episode is defined as the number of (fixed width) windows (over the data) in which it occurs at least once. Recently, it was proposed [16] that the window width can be automatically adjusted by specifying the maximum allowable time separation between events. The windows-based count is not immediately relatable to the more intuitive notion of frequency, namely, the number of occurrences. Another frequency proposed in [1] is the number of minimal occurrences. A minimal occurrence is a window containing the episode such that no proper subwindow of it contains an occurrence of the episode. The method indicated in [1] for counting minimal occurrences is inefficient (with memory needed being of the order of data length). In contrast, the memory needed by the windows-based count is only of the order of the number of nodes in the episode. However, it is shown through simulations in [1] that the minimal occurrences-based count typically runs 30-40 times faster.

In the next section, we present a new definition for episode frequency by restricting the count to certain precisely defined types of occurrences of the episode. The algorithm we present for frequency counting has the same order of space complexity as that of the windows-based frequency (and, as a matter of fact, needs less temporary memory). We show through simulations that it runs much faster.

3 Frequency Counting

Intuitively, the total number of occurrences of an episode seems to be the most natural choice for its frequency. As proposed in [1], the occurrence of episodes in an event sequence may be recognized using finite state automata. For example, for the episode $(A \rightarrow B \rightarrow C)$, there would be an automaton that transits to state "A" on seeing an event of type A and then waits for an event of type B to transit to its next state and so on. When this automaton transits to its final state, an occurrence of the episode is complete. Different instances of the automaton of an episode are needed to keep track of all state transition possibilities and, hence, count all relevant occurrences. Counting *all* occurrences is inefficient and, further, it does not even guarantee that the frequency of a subepisode is greater than or equal to that of the episode.

Each occurrence of an episode is associated with a set of events in the data stream. Two occurrences are said to be distinct if they do not share any events. In the data sequence (1), there are only two distinct occurrences of episode $(A \rightarrow B \rightarrow C)$, though the total number of occurrences is four. While the number of distinct occurrences of an episode seems attractive as a frequency measure, it is difficult to count it efficiently. Consider the sequence

$$\langle (A,1), (B,2), (A,3), (B,4), (A,7), (B,8), \ldots \rangle. \tag{2}$$

In such a case, (in principle) any number of instances of the $(A \rightarrow B \rightarrow C)$ automaton may be needed, all waiting in the second state, since any number of Cs may appear later in the event sequence. Hence, there is a need for further restricting the kind of occurrences to count.

Definition 1. *Two occurrences of an episode are said to be* **nonoverlapping** *if no event associated with one appears in between the events associated with the other. The* **frequency** *of an episode is defined as the maximum number of nonoverlapping occurrences of the episode in the event sequence.*

There are at most two nonoverlapping occurrences of the episode $(A \rightarrow B \rightarrow C)$ in (1): $\{(A,1), (B,3), (C,6)\}$ and $\{(A,14), (B,15), (C,17)\}$. Clearly, the set of nonoverlapping occurrences is a subset of the collection of all possible occurrences. In the event sequence (1), consider the occurrence of $(A \rightarrow B \rightarrow C)$ given by $\{(A,1), (B,15), (C,17)\}$. We do not have any other occurrence of the episode which is nonoverlapping with this occurrence. However, by not considering this but by considering some other occurrences (as above), it is possible to obtain a set of two nonoverlapping occurrences. This is why Definition 1 prescribes the frequency as the *maximum* number of nonoverlapping

occurrences. Counting of nonoverlapping occurrences can be done efficiently. For example, in (2), we need to keep track of only the last pair of event types A and B since all we need to recognize are nonoverlapping occurrences.

3.1 Counting Nonoverlapping Occurrences

This section presents an algorithm that counts the maximum number of nonoverlapping occurrences for each episode in a set of (candidate) episodes. Proceeding left to right along the event sequence and counting the innermost occurrence in any set of overlapping occurrences always yields the largest collection of nonoverlapping occurrences for an episode.

Given a set of candidate episodes $\mathcal{C}$, ALGORITHM I returns the set of frequent episodes $\mathcal{F}$. ALGORITHM I is an automata-based counting scheme similar in spirit to that in [1]. The main data structure here is the $waits(\cdot)$ list. Since there are many candidate episodes, for each of which there are multiple occurrences, at any time there would be many automata waiting for many event types to occur. In order to access the automata efficiently, for each event type A, automata that can now accept A are stored in the list $waits(A)$. This list contains entries of the form (α, j), meaning that an automaton of episode α is waiting for event type A as its jth event. That is, if event type A occurs now in the event sequence, this automaton would accept it and transit to the jth state. (For an episode α, the event type of its jth node is denoted $\alpha[j]$.) At any time (including at the start of the counting process) there would be automata waiting for event types corresponding to the first nodes of all candidate episodes. This is how the $waits(\cdot)$ list is initialized (line 5 in pseudocode). After that, with every event in the data stream, the $waits(\cdot)$ list is appropriately updated. If E_i is the next event in the input sequence, then the $waits(\cdot)$ list is updated as follows: (α, j) is removed from $waits(E_i)$ and $(\alpha, j+1)$ added to $waits(\alpha[j+1])$ (lines 9-15). Since we loop over all elements in $waits(E_i)$, it is inappropriate to add to it from within the loop when $\alpha[j+1] = E_i$. Hence, in such cases, $(\alpha, j+1)$ is added to a temporary list, called bag, which is later emptied into $waits(E_i)$ on coming out of the loop (lines 12-13, 21).

ALGORITHM I, at the instant of completion of an episode's occurrence, resets all automata for that episode (lines 18-20). This ensures that any collection of overlapping occurrences increments the episode frequency by exactly one.

ALGORITHM I: NONOVERLAPPING OCCURRENCE COUNT

Input: Set $\mathcal{C}$ of (candidate) episodes, event stream $s = \langle (E_1, t_1), \ldots, (E_n, t_n)) \rangle$, frequency threshold $\lambda_{\min} \in [0, 1]$

Output: The set $\mathcal{F}$ of frequent episodes in $\mathcal{C}$

1: Initialize $bag = \phi$
2: **for all** event types A **do**
3: Initialize $waits(A) = \phi$
4: **for all** $\alpha \in \mathcal{C}$ **do**
5: Add $(\alpha, 1)$ to $waits(\alpha[1])$
6: Initialize $\alpha.freq = 0$
7: **for** $i = 1$ to n **do**
8: **for all** $(\alpha, j) \in waits(E_i)$ **do**
9: **if** $j \neq 1$ **then**
10: Remove (α, j) from $waits(E_i)$
11: **if** $j < |\alpha|$ **then**
12: **if** $\alpha[j+1] = E_i$ **then**
13: Add $(\alpha, j+1)$ to bag
14: **else**
15: Add $(\alpha, j+1)$ to $waits(\alpha[j+1])$
16: **if** $j = |\alpha|$ **then**
17: Update $\alpha.freq = \alpha.freq + 1$
18: **for all** $1 \leq k < |\alpha|$ **do**
19: Remove $(\alpha, k+1)$ from $waits(\alpha[k+1])$
20: Remove $(\alpha, k+1)$ from bag
21: Empty bag into $waits(E_i)$
22: Output $\mathcal{F} = \{\alpha \in \mathcal{C}$ such that $\alpha.freq \geq n\lambda_{\min}\}$

In this algorithm, the user needs to specify only the frequency threshold. No other parameters such as window width, etc., are needed. At the same time, additional temporal constraints (like those in [1] or [16]) can be readily incorporated. For example, it may be useful to have an expiry time for episode occurrences so that widely spread out events are not regarded as an occurrence. Such conditions may be enforced by testing the appropriate time constraint before permitting a transition into a new state.

3.2 Space and Time Complexity

To count occurrences of an N-node episode α, the algorithm needs N automata. Thus, when counting frequencies of a collection $\mathcal{C}$ of such episodes, the space complexity is $\mathcal{O}(|\mathcal{C}|N)$. This is same as the space complexity of the windows-based count of [1]; however, the temporary memory needed by ALGORITHM I is lesser since there is no need for lists such as the $beginsat(t)$ as was used in [1] (which stores automata initialized at each time instant t) whose size is data length dependent. In addition, the algorithm in [1] needs to update the list to take care of episodes "falling off" the left end of the window. This step is absent in ALGORITHM I.

To determine the time complexity of ALGORITHM I, note that it enters the main loop n times, once for each event in the input sequence. Then, each automaton in the $waits$ list needs to be updated. There are $|\mathcal{C}|$ candidate episodes, each with at most N automata (since they are N-node episodes). Thus, the worst-case time complexity of ALGORITHM I is $\mathcal{O}(|\mathcal{C}|Nn)$. This expression is similar to the worst-case time complexity of the windows-based count as obtained in [1], the only difference being that, in our case, n is the number of events in the input sequence, while, in [1], it is the number of time-ticks spanning the input sequence. In the windows-based count, the automata need to be updated for every shift of the sliding window, while ALGORITHM I does it only each time a new event occurs in the sequence. This is an advantage if the time spanned far exceeds the number of events in the input stream.

4 EPISODE GENERATING HMMS (EGHS)

The most important consequence of the new definition of episode frequency introduced in the previous section is that it allows us to formally connect frequent episode discovery

with learning generative models for the data stream in the form of some specialized Hidden Markov Models (HMMs). We call this class of specialized HMMs Episode Generating HMMs (EGHs). This section defines the class of EGHs and formally proves the connection between frequent episodes and EGHs.

HMMs are very popular for modeling and analysis of time series data with applications in areas ranging from speech processing to bioinformatics. For the sake of completeness, we begin with a brief overview of discrete HMMs [17], [18].

An HMM contains a Markov chain over some state space. The states themselves are unobservable. In each state, the model emits a symbol from a finite symbol set according to a symbol probability distribution. This stream of symbols is the observable output sequence of the model. A discrete HMM is specified by the state space, S, the state transition probability matrix, $\bar{P}$, the initial state probabilities, π and the symbol probability distributions, b. (We use the following notation: For each $i \in S$, π_i is the probability that the initial state is i and $b_i(\cdot)$ is a probability distribution according to which symbols are emitted from state i. The element p_{ij} of matrix $\bar{P}$ denotes the probability of making a transition from state i to state j.) Often, the state space is clear from context and we specify an HMM by the triple $\Lambda = (\pi, \bar{P}, b)$.

Let $\mathbf{o} = (o_1, o_2, \ldots, o_T)$ be an observed symbol sequence. The joint probability of the output sequence $\mathbf{o}$ and a state sequence $\mathbf{q} = (q_1, q_2, \ldots, q_T)$ given an HMM Λ is

$$P(\mathbf{o}, \mathbf{q} \mid \Lambda) = \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^{T} p_{q_{t-1}q_t} b_{q_t}(o_t). \tag{3}$$

The probability of an HMM Λ outputting the sequence $\mathbf{o}$ is obtained by summing the above over all state sequences:

$$P(\mathbf{o} \mid \Lambda) = \sum_{\mathbf{q}} P(\mathbf{o}, \mathbf{q} \mid \Lambda). \tag{4}$$

Often, this data likelihood is assessed by simply evaluating the joint likelihood of (3) only along a *most likely* state sequence, $\mathbf{q}^*$ (which may not be unique), where

$$\begin{aligned} \mathbf{q}^* &= \arg\max_{\mathbf{q}} P(\mathbf{q} \mid \mathbf{o}, \Lambda) \\ &= \arg\max_{\mathbf{q}} P(\mathbf{o}, \mathbf{q} \mid \Lambda). \end{aligned} \tag{5}$$

Note that $\mathbf{q}^*$ is dependent on Λ and we use the notation $\mathbf{q}^*_\Lambda$ when it is necessary to show this dependence explicitly. Given some data, an HMM for the data can be learned through maximum-likelihood estimation by finding a Λ, from among a given class of HMMs, to maximize $P(\mathbf{o} \mid \Lambda)$. In many applications (e.g., speech recognition), one needs to compare the probabilities of different HMMs generating a given sequence $\mathbf{o}$. In such cases, it is often assumed [18, Chapters 25-26] that

$$\mathbf{A1}: \arg\max_{\Lambda} P(\mathbf{o} \mid \Lambda) = \arg\max_{\Lambda} P(\mathbf{o}, \mathbf{q}^*_\Lambda \mid \Lambda),$$

where the maximum is over some set of HMMs of interest. This simplifies computation since we need to compute data likelihood only along the most likely state sequence. We use assumption A1 in some of our analysis to follow.

4.1 Structure of Episode Generating HMMs

We now introduce our specialized HMMs, namely, the Episode Generating HMMs (EGHs). Since we want to connect episodes with HMMs, the symbol set would be the set of event types and the output sequences would be the event sequences.[1]

In an Episode Generating HMM (EGH), the number of states is always even and the state space is denoted by $\mathcal{S} = \{1, \ldots, N, N+1, \ldots, 2N\}$. This state space is partitioned into two: the episode states, $\mathcal{S}_e = \{1, \ldots, N\}$, and the noise states, $\mathcal{S}_n = \{N+1, \ldots, 2N\}$. The symbol set would be the set of event types. The symbol probability distribution is the uniform distribution for all noise states and a delta function for each of the episode states. (That is, in each episode, state one of the symbols is emitted with probability 1.) We specify all the episode state symbol probability distributions by $\mathcal{A} = (A_1, \ldots, A_N)$, where the notation is that episode state i can only emit the symbol (or event of type) A_i. The state transition probabilities have a restricted structure. All transitions into noise states have probability η and all transitions into episode states have probabilities $(1-\eta)$, where $\eta \in (0, 1]$ is called the noise parameter. An episode state k can transit into either the noise state $(N+k)$ with probability η or the episode state $(k \bmod N) + 1$ with probability $(1-\eta)$. A noise state $(N+k)$ can either remain there with probability η or transit to the episode state $(k \bmod N) + 1$ with probability $(1-\eta)$. The initial state is 1 with probability $(1-\eta)$ and $2N$ with probability η. Thus, all transition and initial state probabilities are determined by a single parameter, namely, η.

An EGH is specified by $\Lambda = (\mathcal{S}, \mathcal{A}, \eta)$. $\mathcal{A} = (A_1, \ldots, A_N)$ is referred to as the episode state symbol parameters and η as the noise parameter. $\mathcal{E}$ denotes the class of all EGHs.

An example EGH with six states is shown in Fig. 1. The dotted circle labeled 0, along with its outgoing arcs, represents the initial state probability distribution. The symbol probability distributions are shown alongside corresponding nodes. As per our notation, states 1, 2, 3 are episode states and states 4, 5, 6 are noise states. This example EGH can only emit an "A" in state 1, a "B" in state 2, and a "C" in state 3. It is easy to see that (if η is small) the output sequence generated by this example EGH would be an event sequence with many occurrences of episode $(A \rightarrow B \rightarrow C)$. Thus, the EGH in Fig. 1 is what we would like to associate with that episode, as will be clear from the discussion to follow.

4.2 Class of EGHs with a Fixed η

We begin our investigation of a formal connection between frequent episodes and EGHs by first considering a subclass of EGHs with a fixed η.

Definition 2. *The subclass $\mathcal{E}_\eta$ is defined as the collection of all EGHs (out of $\mathcal{E}$) with a fixed noise parameter η.*

1. Though each event in the event sequence is specified by an event type as well as a time of occurrence, the actual values of event times are not important. The event times are used only to order events and only this ordering is needed to count episode occurrences. Thus, when analyzing the frequent episode discovery process, it is enough to consider a model which generates an ordered sequence of event types.

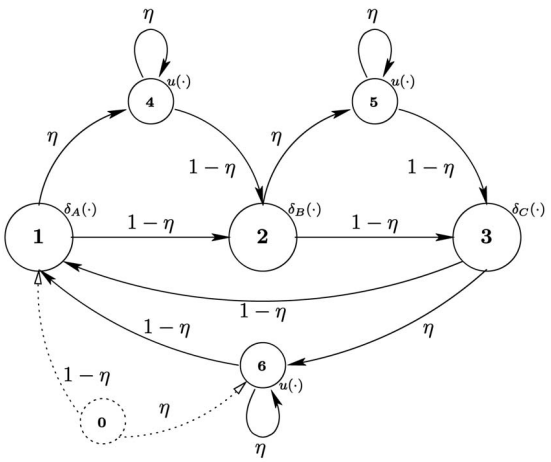

Fig. 1. An example EGH. Symbol probabilities are shown alongside corresponding nodes. $\delta_A(\cdot)$ denotes a pdf with probability 1 for symbol A and 0 for all others and, similarly, for $\delta_B(\cdot)$ and $\delta_C(\cdot)$. $u(\cdot)$, denotes the uniform pdf over the symbol set.

Definition 3. *The EGH associated with episode* $\alpha = (A_1 \rightarrow \cdots \rightarrow A_N)$ *is* $\Lambda_\alpha = (\mathcal{S}, \mathcal{A}, \eta) \in \mathcal{E}_\eta$, *where* $\mathcal{S} = \{1, \ldots, 2N\}$ *and* $\mathcal{A} = (A_1, \ldots, A_N)$.

This episode-EGH association has an important property which is stated below in *Theorem 1*.

Theorem 1. *Consider the class* $\mathcal{E}_\eta$, *of EGHs with* $\eta < \frac{M}{M+1}$ *(where* M *is the cardinality of the symbol set). Let* $\mathbf{o} = (o_1, \ldots, o_T)$ *be the given event sequence. Let* α *and* β *be two* N*-node episodes occurring in* $\mathbf{o}$. *Let* Λ_α *and* Λ_β *be the EGHs from* $\mathcal{E}_\eta$ *associated with* α *and* β *according to Definition 3. Let* f_α *and* f_β *denote the frequencies of* α *and* β, *respectively (where frequency is the maximum number of nonoverlapping occurrences, as in Definition 1). Let* $\mathbf{q}^*_\alpha$ *and* $\mathbf{q}^*_\beta$ *be the most likely state sequences for* $\mathbf{o}$ *under* Λ_α *and* Λ_β, *respectively. Then, 1)* $f_\alpha > f_\beta$ *implies* $P(\mathbf{o}, \mathbf{q}^*_\alpha \mid \Lambda_\alpha) > P(\mathbf{o}, \mathbf{q}^*_\beta \mid \Lambda_\beta)$, *and 2)* $P(\mathbf{o}, \mathbf{q}^*_\alpha \mid \Lambda_\alpha) > P(\mathbf{o}, \mathbf{q}^*_\beta \mid \Lambda_\beta)$ *implies* $f_\alpha \geq f_\beta$.

Proof. Since the state space of an EGH is partitioned as $\mathcal{S} = \mathcal{S}_e + \mathcal{S}_n$, given any state sequence $\mathbf{q}$, we can decompose it into two subsequences, $\mathbf{q}_e$ and $\mathbf{q}_n$, each of which consist of elements of $\mathbf{q}$ that are in $\mathcal{S}_e$ and $\mathcal{S}_n$, respectively.

For any HMM, the joint probability of an observation sequence and a state sequence is given by (3). For an EGH, this expression is particularly simple. Each transition probability is either η or $(1-\eta)$. Whenever the transition probability $p_{q_{t-1}q_t}$ is $(1-\eta)$, the state q_t has to be an episode state and, hence, the corresponding $b_{q_t}(o_t)$ is either 1 or 0, depending on whether or not the symbol o_t can be emitted in the episode state q_t. Similarly, whenever $p_{q_{t-1}q_t}$ is η, the corresponding $b_{q_t}(o_t)$ is $(\frac{1}{M})$. The same thing is true for π_{q_1} and $b_{q_1}(o_1)$. Thus, for any state sequence such that the joint probability is nonzero and for any EGH Λ, we have

$$P(\mathbf{o}, \mathbf{q} \mid \Lambda) = \left(\frac{\eta}{M}\right)^{|\mathbf{q}_n|} (1-\eta)^{|\mathbf{q}_e|} \tag{6}$$

$$= \left(\frac{\eta}{M}\right)^{T} \left(\frac{1-\eta}{\eta/M}\right)^{|\mathbf{q}_e|}. \tag{7}$$

Here, $|\mathbf{q}_n|$ and $|\mathbf{q}_e|$ denote lengths of the respective subsequences and we have used the fact that $|\mathbf{q}_n| + |\mathbf{q}_e| = |\mathbf{q}| = T$, the length of the output (or event) sequence. Under the restriction $\eta < (\frac{M}{M+1})$, we have $(\frac{1-\eta}{\eta/M}) > 1$. Hence, from (7), $P(\mathbf{o}, \mathbf{q} \mid \Lambda)$ is monotonically increasing with $|\mathbf{q}_e|$. Now, the most likely state sequence, $\mathbf{q}^*$, is given by

$$\mathbf{q}^* = \arg\max_{\mathbf{q}} \left(\frac{\eta}{M}\right)^{T} \left(\frac{1-\eta}{\eta/M}\right)^{|\mathbf{q}_e|}, \tag{8}$$

$$= \arg\max_{\mathbf{q}} |\mathbf{q}_e|. \tag{9}$$

Thus, any most likely state sequence is one that spends the longest time in episode states. While $\mathbf{q}^*$ does not explicitly depend on η (as long as $\eta < \frac{M}{M+1}$), it very much depends on the other parameters, $\mathcal{S}$ and $\mathcal{A}$, of model Λ (although the notation does not explicitly show this). Now, we have

$$P(\mathbf{o}, \mathbf{q}^* \mid \Lambda) = \left(\frac{\eta}{M}\right)^{T} \left(\frac{1-\eta}{\eta/M}\right)^{|\mathbf{q}^*_e|}, \tag{10}$$

where $\mathbf{q}^*_e$ denotes the subsequence of episode states in a most likely state sequence, $\mathbf{q}^*$.

Equations (9) and (10) are true for any EGH Λ with $\eta < (\frac{M}{M+1})$. Now, consider episodes, α and β, and their associated EGHs, Λ_α and Λ_β. Let $\mathbf{q}^*_{\alpha e}$ denote the subsequence of episode states of a most likely state sequence for EGH Λ_α and, similarly, $\mathbf{q}^*_{\beta e}$. Given (10), proof of the first part of Theorem 1 is now complete if we can show that $f_\alpha > f_\beta$ implies $|\mathbf{q}^*_{\alpha e}| > |\mathbf{q}^*_{\beta e}|$.

Due to the constraints imposed on the state transition structure, in any state sequence (that has nonzero probability) of EGH Λ_α, episode states have to occur in the same sequence as the event types in episode α. Also, the episode state corresponding to the first event can appear a second time in a state sequence only if one full cycle of all episode states has already appeared in this state sequence. Thus, if there are at least two nonoverlapping occurrences of the N-node episode, α, in $\mathbf{o}$, then there is a state sequence having nonzero probability for the EGH Λ_α with at least $2N$ episode states in it. Similarly, given a state sequence (with positive probability) that has $2N$ or more episode states, there are at least two nonoverlapping occurrences of the episode α. The reason for this is as follows: By the allowed transitions in an EGH, $2N$ episode states can come about only by having visited the episode states 1 to N, in that same order, twice. Thus, the events in the event sequence corresponding to the positions of the $2N$ episode states in the state sequence constitute two nonoverlapping occurrences of the episode. As was seen earlier, $\mathbf{q}^*_\alpha$, a most likely state sequence of $\Lambda_\alpha \in \mathcal{E}_\eta$, is one that has the maximum possible number of episode states in it and the number of episode states in it is $|\mathbf{q}^*_{\alpha e}|$. By definition, f_α is the maximum possible number of nonoverlapping occurrences of α in $\mathbf{o}$. Hence, $|\mathbf{q}^*_{\alpha e}| \geq N f_\alpha$. By how much can $|\mathbf{q}^*_{\alpha e}|$ exceed $N f_\alpha$? The difference has to be less than N since, otherwise, the extra N episode states in this state sequence would constitute another occurrence of the

episode which is nonoverlapping with all others; but, it is given that f_α is the maximum number of nonoverlapping occurrences. Putting together all these (and noting that what is said about α is also true for β), we get

$$\begin{array}{rcl} Nf_\alpha \le |\mathbf{q}^*_{\alpha e}| & \le & Nf_\alpha + N - 1 \\ Nf_\beta \le |\mathbf{q}^*_{\beta e}| & \le & Nf_\beta + N - 1. \end{array} \quad (11)$$

Since f_α and f_β are integers, $f_\alpha > f_\beta$ implies $f_\beta \le (f_\alpha - 1)$. Now, using (11), we have

$$|\mathbf{q}^*_{\beta e}| \le N(f_\alpha - 1) + N - 1 = Nf_\alpha - 1 < |\mathbf{q}^*_{\alpha e}|.$$

This shows that $f_\alpha > f_\beta$ implies $|\mathbf{q}^*_{\alpha e}| > |\mathbf{q}^*_{\beta e}|$ and, hence, proof for the first part of Theorem 1 is complete.

For the second part, first note from (10) that $P(\mathbf{o}, \mathbf{q}^*_\alpha \mid \Lambda_\alpha) > P(\mathbf{o}, \mathbf{q}^*_\beta \mid \Lambda_\beta)$ implies $|\mathbf{q}^*_{\alpha e}| > |\mathbf{q}^*_{\beta e}|$. Using (11) (which is true for any pair of episodes), we have

$$Nf_\beta \le |\mathbf{q}^*_{\beta e}| < |\mathbf{q}^*_{\alpha e}| \le Nf_\alpha + N - 1,$$

so that $f_\beta < (f_\alpha + \frac{N-1}{N})$, implying that $f_\beta \le f_\alpha$ since both of these have to be integers. This proves the second part of the theorem. Proof of Theorem 1 is hence complete.□

A consequence of the above result is that, under assumption A1, if α is the most frequent episode, then the probability of the data stream $\mathbf{o}$ under $\Lambda_\alpha \in \mathcal{E}_\eta$ is greater than that under any other EGH in the class $\mathcal{E}_\eta$ (so long as $\eta < \frac{M}{M+1}$). This straightaway gives us our next theorem.

Theorem 2. *Given a data stream* $\mathbf{o}$*, under assumption A1, the maximum-likelihood estimate for a* $2N$-state *EGH over the class of EGHs* $\mathcal{E}_\eta$ *for any* $\eta < (\frac{M}{M+1})$ *is the EGH corresponding to the most frequent* N-node *episode.*

Remark 1. Under assumption A1, given a class of EGHs, the Λ that maximizes $P(\mathbf{o} \mid \Lambda)$ also maximizes $P(\mathbf{o}, \mathbf{q}^*_\Lambda \mid \Lambda)$. This extra assumption gives us Theorem 2 from Theorem 1. As stated earlier, assumption A1 is fairly reasonable and is often used in applications of HMMs. Note that the assumption does not need $P(\mathbf{o} \mid \Lambda) \approx P(\mathbf{o}, \mathbf{q}^*_\Lambda \mid \Lambda)$. Now, suppose that A1 holds when the maximum is taken over any arbitrary set of HMMs. This gives us a stronger version of Theorem 1, namely, that for any two episodes $f_\alpha > f_\beta$ implies $P(\mathbf{o} \mid \Lambda_\alpha) \ge P(\mathbf{o} \mid \Lambda_\beta)$.

Before moving on, we would like to make some important observations regarding the upper bound on η in Theorems 1 and 2. This upper bound, $(\frac{M}{M+1})$, is essentially due to the structure of EGHs. For $\eta < (\frac{M}{M+1})$, the $P(\mathbf{o}, \mathbf{q} \mid \Lambda)$ in (7) monotonically increases with $|\mathbf{q}_e|$. If, instead, $\eta > (\frac{M}{M+1})$, then $P(\mathbf{o}, \mathbf{q} \mid \Lambda)$ will monotonically decrease with $|\mathbf{q}_e|$ (and increase with $|\mathbf{q}_n|$). Further, unlike episode states, noise states can emit all symbols with equal probability. Thus, the EGH can remain in a single noise state for the entire duration of the symbol sequence and this will indeed be the most likely state sequence for the $\eta > \frac{M}{M+1}$ case. When $\eta = (\frac{M}{M+1})$, all state sequences are equally likely. Thus, we have the following.

Remark 2. A most likely state sequence, $\mathbf{q}^*$, for a $2N$-node EGH $\Lambda = (\mathcal{S}, \mathcal{A}, \eta)$ (on data stream $\mathbf{o}$) is 1) one that spends the most time in episode states, if $\eta < (\frac{M}{M+1})$, and 2) one that spends all time in the noise state $2N$, if $\eta \ge (\frac{M}{M+1})$.

4.3 The Final Correspondence between Episodes and EGHs

Consider an episode $\alpha = (A_1 \to \cdots \to A_N)$. In Section 4.2, this episode was associated with the EGH $(\mathcal{S}, \mathcal{A}, \eta)$, in $\mathcal{E}_\eta$, where $\mathcal{A} = (A_1, \ldots, A_N)$ and η was an arbitrary constant. Theorems 1 and 2 provide the basis for such an association (so long as $\eta < \frac{M}{M+1}$). Now, we consider the full class of EGHs $\mathcal{E}$ and seek a unique EGH here to associate with α. For this, another subclass of $\mathcal{E}$ needs to be defined.

Definition 4. *Given an episode* $\alpha = (A_1 \to \cdots \to A_N)$*, subclass* $\mathcal{E}(\alpha)$ *is defined as the collection of EGHs in* $\mathcal{E}$ *of the form* $(\mathcal{S}, \mathcal{A}_\alpha, \eta)$*, where* $\mathcal{S} = \{1, \ldots, 2N\}$*,* $\mathcal{A}_\alpha = (A_1, \ldots, A_N)$*, and* $\eta \in (0, 1]$.

The only candidates in $\mathcal{E}$ that can be meaningfully associated with an episode α are the EGHs in $\mathcal{E}(\alpha)$. In $\mathcal{E}(\alpha)$, there are infinitely many EGHs since η takes all values in $(0, 1]$. Which of these would be a useful episode-EGH association? It is reasonable to associate the episode α with the EGH $(\mathcal{S}, \mathcal{A}_\alpha, \widehat{\eta}_\alpha)$ such that $\widehat{\eta}_\alpha$ maximizes the probability of generating the given event sequence. From Remark 2, the joint probability of $\mathbf{o}$ and a most likely state sequence for EGH $\Lambda = (\mathcal{S}, \mathcal{A}_\alpha, \eta) \in \mathcal{E}(\alpha)$ is

$$P(\mathbf{o}, \mathbf{q}^* \mid \Lambda) = \begin{cases} \left(\frac{\eta}{M}\right)^T \left(\frac{1-\eta}{\eta/M}\right)^{K_{\alpha e}(\mathbf{o})} & \text{if } 0 < \eta < \frac{M}{M+1} \\ \left(\frac{\eta}{M}\right)^T & \text{if } \frac{M}{M+1} \le \eta \le 1, \end{cases} \quad (12)$$

where $K_{\alpha e}(\mathbf{o})$ denotes the maximum possible number of episode states in a state sequence with nonzero probability when generating $\mathbf{o}$ from an EGH with episode state symbol parameters $\mathcal{A}_\alpha$.

Remark 3. For an EGH with $\eta < (\frac{M}{M+1})$, it is easy to see that $|\mathbf{q}^*_{\alpha e}| = K_{\alpha e}(\mathbf{o})$ and this new notation emphasizes the dependence on $\mathbf{o}$. However, if $\eta \ge (\frac{M}{M+1})$, the optimal state sequence is comprised of only noise states and, hence, $|\mathbf{q}^*_{\alpha e}| = 0$. Even in such cases, $K_{\alpha e}(\mathbf{o})$ is well-defined. Further, from the proof of Theorem 1, it is clear that we would always have $(Nf_\alpha) \le K_{\alpha e}(\mathbf{o}) \le (Nf_\alpha + N - 1)$, where f_α is the frequency of episode α (even if the state sequence with $K_{\alpha e}(\mathbf{o})$ episode states is *not* a most likely state sequence).

To find the best η, we need to find the maxima of both expressions on the RHS of (12) and then compare them. It is clear by inspection that the second expression is maximized at $\eta = 1$ and the maximum value is $(\frac{1}{M})^T$. If $P(\mathbf{o}, \mathbf{q}^* \mid \Lambda)$ given by (12) is maximized at $\eta = 1$, then the best EGH association for α would be one with $\eta = 1$, which is merely a random iid model. We discuss this case later on. For now, let us assume that η that maximizes $P(\mathbf{o}, \mathbf{q}^* \mid \Lambda)$ comes from the maximizer of the first expression in the RHS of (12). This expression is concave in η over $(0, 1]$ and its partial derivative (with respect to η) is

$$M^{(K_{\alpha e}(\mathbf{o})-T)}\eta^T\left(\frac{1-\eta}{\eta}\right)^{K_{\alpha e}(\mathbf{o})}\left[\frac{T}{\eta}-\frac{K_{\alpha e}(\mathbf{o})}{\eta(1-\eta)}\right].$$

By equating this expression to zero, it is easily seen that the *unique* maximizer is

$$\widehat{\eta}_\alpha = \left(\frac{T-K_{\alpha e}(\mathbf{o})}{T}\right). \tag{13}$$

This $\widehat{\eta}_\alpha$ value is used in our final episode-EGH association. Note that, depending on how large or small $K_{\alpha e}(\mathbf{o})$ is, $\widehat{\eta}_\alpha$ may or may not belong to the interval $(0, \frac{M}{M+1})$.

Definition 5. *The EGH associated with episode $\alpha = (A_1 \rightarrow \cdots \rightarrow A_N)$ is $\Lambda_\alpha = (\mathcal{S}, \mathcal{A}_\alpha, \eta_\alpha)$, where $\mathcal{S} = \{1, \ldots, 2N\}$, $\mathcal{A}_\alpha = (A_1, \ldots, A_N)$, and η_α is set equal to $(\frac{T-K_{\alpha e}(\mathbf{o})}{T})$ if it is less than $\frac{M}{M+1}$ and to 1 otherwise.*

Earlier, Theorem 1 established that, when associating an episode with an EGH from $\mathcal{E}_\eta$, the more frequent episodes are associated with EGHs with higher likelihoods. Since, in our final association, η is no longer a fixed quantity across episodes (and is, in fact, data dependent as well), it is relevant to ask whether the new association preserves such ordering. The theorem below states that the answer to this question is yes, thereby justifying this choice of episode-EGH association.

Theorem 3. *Let $\mathbf{o} = (o_1, \ldots, o_T)$ be the given event sequence. Let α and β be two N-node episodes occurring in $\mathbf{o}$ with frequencies f_α and f_β, respectively. Let Λ_α and Λ_β be the EGHs associated with α and β according to Definition 5. Let $\mathbf{q}^*_\alpha$ and $\mathbf{q}^*_\beta$ be most likely state sequences for $\mathbf{o}$ under Λ_α and Λ_β, respectively. If η_α and η_β are both less than $\frac{M}{M+1}$, then 1) $f_\alpha > f_\beta$ implies $P(\mathbf{o}, \mathbf{q}^*_\alpha \mid \Lambda_\alpha) > P(\mathbf{o}, \mathbf{q}^*_\beta \mid \Lambda_\beta)$ and 2) $P(\mathbf{o}, \mathbf{q}^*_\alpha \mid \Lambda_\alpha) > P(\mathbf{o}, \mathbf{q}^*_\beta \mid \Lambda_\beta)$ implies $f_\alpha \geq f_\beta$.*

Proof. Since $\eta_\alpha < (\frac{M}{M+1})$, from (12), we have

$$P(\mathbf{o}, \mathbf{q}^*_\alpha \mid \Lambda_\alpha) = \left(\frac{\eta_\alpha}{M}\right)^T\left(\frac{1-\eta_\alpha}{\eta_\alpha/M}\right)^{K_{\alpha e}(\mathbf{o})}. \tag{14}$$

A similar expression holds for $P(\mathbf{o}, \mathbf{q}^*_\beta \mid \Lambda_\beta)$, too, because we are given $\eta_\beta < (\frac{M}{M+1})$. Since η_α is the unique maximizer of the first expression on the RHS of (12), we have

$$\left(\frac{\eta_\alpha}{M}\right)^T\left(\frac{1-\eta_\alpha}{\eta_\alpha/M}\right)^{K_{\alpha e}(\mathbf{o})} > \left(\frac{\eta_\beta}{M}\right)^T\left(\frac{1-\eta_\beta}{\eta_\beta/M}\right)^{K_{\alpha e}(\mathbf{o})}. \tag{15}$$

For the first part of Theorem 3, note that $K_{\alpha e}(\mathbf{o}) > K_{\beta e}(\mathbf{o})$ because it is given that $f_\alpha > f_\beta$. Since $\eta_\beta < \frac{M}{M+1}$ (which means $\frac{1-\eta_\beta}{\eta_\beta/M} > 1$) and since $K_{\alpha e}(\mathbf{o}) > K_{\beta e}(\mathbf{o})$, we have

$$\left(\frac{\eta_\beta}{M}\right)^T\left(\frac{1-\eta_\beta}{\eta_\beta/M}\right)^{K_{\alpha e}(\mathbf{o})} > \left(\frac{\eta_\beta}{M}\right)^T\left(\frac{1-\eta_\beta}{\eta_\beta/M}\right)^{K_{\beta e}(\mathbf{o})}. \tag{16}$$

Now, from (15)-(16) and (14), it follows that,

$$P(\mathbf{o}, \mathbf{q}^*_\alpha \mid \Lambda_\alpha) > P(\mathbf{o}, \mathbf{q}^*_\beta \mid \Lambda_\beta),$$

which proves the first part of Theorem 3.

For the second part of Theorem 3, we have

$$\begin{aligned}\left(\frac{\eta_\alpha}{M}\right)^T\left(\frac{1-\eta_\alpha}{\eta_\alpha/M}\right)^{K_{\alpha e}(\mathbf{o})} &> \left(\frac{\eta_\beta}{M}\right)^T\left(\frac{1-\eta_\beta}{\eta_\beta/M}\right)^{K_{\beta e}(\mathbf{o})} \\ &> \left(\frac{\eta_\alpha}{M}\right)^T\left(\frac{1-\eta_\alpha}{\eta_\alpha/M}\right)^{K_{\beta e}(\mathbf{o})},\end{aligned} \tag{17}$$

where the first inequality comes directly from $P(\mathbf{o}, \mathbf{q}^*_\alpha \mid \Lambda_\alpha) > P(\mathbf{o}, \mathbf{q}^*_\beta \mid \Lambda_\beta)$ (which is the condition given) and the second one comes from the fact that η_β is the unique maximizer of the first expression on the RHS of (12) for $P(\mathbf{o}, \mathbf{q}^*_\beta \mid \Lambda_\beta)$. Equation (17) implies $K_{\alpha e}(\mathbf{o}) > K_{\beta e}(\mathbf{o})$, which, from Remark 3, implies $f_\alpha \geq f_\beta$. This completes the proof of Theorem 3. □

5 Consequences of the Episode-EGH Connection

The theorems proven in the previous section provide a formal connection between frequent episodes and generative models in the form of EGHs. We now explore a few consequences of this formal connection.

5.1 Learning Generative Models

Occurrences of an episode in an event stream are like certain kinds of substrings. An occurrence of $(A \rightarrow B \rightarrow C)$ in $\mathbf{o}$ is like a substring Aw_1Bw_2C, where w_1, w_2 are any strings of event types. Different occurrences of the episode may involve different w_1, w_2. Thus, EGHs capture the basic idea of episodes in event streams and, hence, are a good class of generative models to consider.

Let $\mathbf{o}$ be the given event sequence of length T and let α be an N-node episode whose frequency is f_α. We associate episode α with EGH Λ_α from class $\mathcal{E}$ as given by Definition 5. The noise parameter of this EGH is given by $\eta_\alpha = \left(\frac{T-K_{\alpha e}(\mathbf{o})}{T}\right)$ if it is less than $\frac{M}{M+1}$ and by $\eta_\alpha = 1$ otherwise. Now, let α be the most frequent N-node episode in the given data stream $\mathbf{o}$. Then, EGH Λ_α would be a maximum-likelihood estimate for a $2N$-state EGH over the class $\mathcal{E}$ of EGHs if $\widehat{\eta}_\alpha < \frac{M}{M+1}$ and $P(\mathbf{o}, \mathbf{q}^* \mid \Lambda_\alpha) > \left(\frac{1}{M}\right)^T$. Otherwise, the maximum-likelihood estimate is an EGH with noise parameter $\eta = 1$ (which is equivalent to a uniformly distributed iid model). Given an episode α, we only need the value of η_α to obtain the associated EGH and this can be obtained from the frequency counting algorithm. From Remark 3, it is clear that Nf_α is a very good approximation for $K_{\alpha e}(\mathbf{o})$. Notwithstanding this, the exact value of $K_{\alpha e}(\mathbf{o})$ is also easily obtainable as a byproduct of ALGORITHM 1, by adding to Nf_α, the number of state transitions of the longest partial occurrence of α available on reaching the end of the event sequence. Thus, using only the output of our frequency counting algorithm, we can obtain a reasonable generative model for the data stream.

Another interesting consequence of the results proven here is the possibility of deriving a generative model for the data in terms of a mixture of our specialized HMMs. In general, given a data stream, estimating a mixture of HMMs is a hard problem. Our theoretical connection can help get

to such a model. Consider $\mathbf{q}^*_\alpha$, a most likely state sequence of Λ_α, the EGH corresponding to the most frequent episode α. The subsequence of episode states, $\mathbf{q}^*_{\alpha e}$, indicates which part of the data is "explained" by episode α. Hence, one can intuitively say that, to get a good mixture model, we need to consider next not the next most frequent episode but that frequent episode β whose $\mathbf{q}^*_{\beta e}$ has the least intersection with $\mathbf{q}^*_{\alpha e}$. Such a procedure can yield a subset of frequent episodes which together best explain the data. This can be thought of as a way to partition the data into substreams, each having different models. The formal connection proven here can help develop such mixture models for data.

5.2 Significant Frequent Episodes

Our formal connection between episodes and EGHs also gives rise to some interesting ideas regarding assessing the significance of frequent episodes discovered.

There have been some recent results regarding assessing the significance of frequent episodes in event sequences [19], [20], [21]. Here, it is shown that, using the central limit theorem, one can bound the probability that the frequency of an episode is above or below some threshold given a generative model for the data. Then, one can first use some training data to estimate this model (assuming either independence or Markovian dependence) and then assess the significance of an episode (in another data stream from the same source). In [22], similar probabilistic techniques are used in the context of counting all occurrences of a string with wild cards.

In [19], [20], [21], the frequency measure used is the windows-based frequency of [1] and it appears difficult to extend it to the case of nonoverlapping occurrences of an episode. Moreover, we take a somewhat different view of assessing the significance of episodes whereby we do not need any training data to estimate a model for the data generation process. Instead, the connection between episodes and EGHs is used to assess the significance of episodes as explained below.

Consider the case when $\widehat{\eta}_\alpha > (\frac{M}{M+1})$. This means that $P(\mathbf{o}, \mathbf{q}^* \mid \Lambda)$ is maximized (over $\mathcal{E}$) at $\eta = 1$. That is, the likelihood of any EGH with $\eta < 1$ generating the data is less than that of an iid model having generated the data stream $\mathbf{o}$. As was noted earlier, $\widehat{\eta}_\alpha > (\frac{M}{M+1})$ means f_α is below some threshold. Thus, we can test whether an episode is "insignificant" based on its frequency. We put this in a formal hypothesis testing framework below.

Consider an event sequence $\mathbf{o}$ and an episode α. Definition 5 associates the EGH $\Lambda_\alpha = (\mathcal{S}, \mathcal{A}_\alpha, \eta_\alpha)$ with α, where $\eta_\alpha = (\frac{T - K_{\alpha e}(\mathbf{o})}{T})$ if it is less than $(\frac{M}{M+1})$ and $\eta_\alpha = 1$, otherwise. We now wish to test the (alternate) hypothesis H_1: $\mathbf{o}$ is drawn from the EGH Λ_α, against the (null) hypothesis H_0: $\mathbf{o}$ is drawn from an iid source.

The *likelihood ratio test* rejects the null hypothesis H_0 if

$$L(\mathbf{o}) = \frac{P[\mathbf{o}\ ;\ H_1]}{P[\mathbf{o}\ ;\ H_0]} > \gamma, \tag{18}$$

where γ is a positive threshold obtained by fixing the probability of Type I error (i.e., the probability of wrong rejection of the null hypothesis). $L(\mathbf{o})$ is called the likelihood ratio for $\mathbf{o}$. Since $P[\mathbf{o}; H_1] > P[\mathbf{o}, \mathbf{q}^*; H_1]$, the test:

$$\text{if } L_1(\mathbf{o}) = \frac{P[\mathbf{o}, \mathbf{q}^*\ ;\ H_1]}{P[\mathbf{o}\ ;\ H_0]} > \gamma, \quad \text{reject } H_0, \tag{19}$$

would be more conservative in accepting an episode as significant and, so, we consider $L_1(\mathbf{o})$ as the test statistic.

If $\eta_\alpha = 1$, then the test statistic is unity and the alternate and null hypotheses are the same. Hence, we essentially need to consider the case when $\eta_\alpha = (\frac{T - K_{\alpha e}(\mathbf{o})}{T}) < (\frac{M}{M+1})$. In this case, given any positive γ, there exists a Γ such that having $[P[\mathbf{o}, \mathbf{q}^*; H_1] > \gamma]$ is equivalent to having $[K_{\alpha e}(\mathbf{o}) > \Gamma]$ (cf. (12)). Thus, $K_{\alpha e}(\mathbf{o})$ is an equivalent test statistic and we reformulate our test as:

$$\text{if } L_2(\mathbf{o}) = K_{\alpha e}(\mathbf{o}) > \Gamma, \quad \text{reject } H_0. \tag{20}$$

The value of Γ to be used in the test is decided by fixing the probability of Type I error. Since the likelihood under H_0 of any sequence of length T is $(\frac{1}{M})^T$, the Type I error probability, denoted P_{FA}, is given by

$$P_{FA} = P[L_2(\mathbf{o}) > \Gamma\ ;\ H_0], \tag{21}$$

$$= \left(\frac{1}{M}\right)^T Q(\Gamma), \tag{22}$$

where $Q(\Gamma)$ denotes the number of sequences of length T (over the M-size alphabet) for which $L_2(\mathbf{o}) > \Gamma$, i.e.,

$$Q(\Gamma) = |\{\mathbf{o}\ ;\ K_{\alpha e}(\mathbf{o}) > \Gamma\}|. \tag{23}$$

Now, the number of T-length sequences (over the M-size alphabet) for which a most likely state sequence (for a given EGH Λ_α) spends *exactly* k instants in episode states is bounded above by $\binom{T}{k}(M-1)^{T-k}$. This is because, there are $\binom{T}{k}$ ways of choosing the k episode state positions and the remaining $(T-k)$ positions can be filled with any of $(M-1)$ symbols (i.e., any symbol except the one that may force an earlier transition to the next episode state). This idea, together with (23), yields an upper bound for $Q(\Gamma)$ which, in turn, gives an upper bound for P_{FA}. Thus, we have

$$P_{FA} \leq \left(\frac{1}{M}\right)^T \sum_{k>\Gamma} \binom{T}{k} (M-1)^{T-k}, \tag{24}$$

$$= 1 - \sum_{k \leq \Gamma} \binom{T}{k} \left(\frac{1}{M}\right)^k \left(1 - \frac{1}{M}\right)^{T-k}, \tag{25}$$

$$\approx 1 - \Phi\left(\frac{\Gamma - \frac{T}{M}}{\sqrt{T(\frac{1}{M})(1-\frac{1}{M})}}\right), \tag{26}$$

where $\Phi(\cdot)$ is the cumulative distribution function (cdf) of a standard normal random variable and the last approximation holds for large T due to the central limit theorem.

Now, we summarize the procedure for testing the significance of an episode α. Fix a level for the test by choosing some upper bound, say ϵ, on Type I error

TABLE 1
Comparison of Γ Values for Various Type I Error Probability Bounds, ϵ, against the Fraction $(\frac{T}{M})$

	ϵ	Γ	% diff. of Γ from T/M
M=25,	0.5	2000	0.0%
T=50000,	0.01	2102	5.1%
T/M=2000	0.001	2145	7.3%
M=50,	0.5	1000	0.0%
T=50000,	0.01	1072	7.2%
T/M=1000	0.001	1097	9.7%

probability, P_{FA}. Since T and M are known, using (26) and the standard normal tables, Γ is fixed as follows:

$$\Gamma = \frac{T}{M} + \sqrt{\left(\frac{T}{M}\right)\left(1 - \frac{1}{M}\right)}\Phi^{-1}(1-\epsilon). \tag{27}$$

Then, reject the null hypothesis (that is, declare α as significant) if $K_{\alpha\epsilon}(\mathbf{o}) > \Gamma$. As stated earlier, Nf_α is a good estimate of $K_{\alpha\epsilon}(\mathbf{o})$ and, if need be, we can find its exact value at the end of the frequency counting algorithm. Table 1 lists Γ for some example T, M, and ϵ values.

Observe that there is no need to explicitly verify whether $\widehat{\eta}_\alpha$ (cf. (13)) is less than $\frac{M}{M+1}$ before testing for the significance of episode α. We simply compute $K_{\alpha\epsilon}(\mathbf{o})$ (or Nf_α) by the frequency counting algorithm and do the test as above. From Table 1 and (27), it can be seen that, when the bound on Type I error probability is 0.5, $\Gamma = \frac{T}{M}$ and Γ increases for lower ϵ values. For any episode α, $\widehat{\eta}_\alpha \geq \frac{M}{M+1}$ would imply $K_{\alpha\epsilon}(\mathbf{o}) \leq \frac{T}{M+1}$ and, hence, α would automatically be declared *not* significant by our test (under the very reasonable assumption that the desired Type I error probability is less than 0.5).

Note that there is no need to estimate any state sequences of HMMs, etc., here. The structure of EGHs and the proofs provided are needed only as a theoretical backbone. Due to this theoretical development, there is now a method to assess the statistical significance of frequent episodes discovered.

5.3 Parameterless Data Mining

Recall from Section 3 that the only parameter needed for our frequency counting algorithm is the frequency threshold. The hypothesis testing framework developed in Section 5.2 allows us to fix this frequency threshold also automatically.

From (27), since T is usually much larger than M, $\frac{T}{M}$ is the dominant factor in the value of Γ. This is also seen from Table 1. This means $\frac{T}{NM}$ is, in general, a good frequency threshold to use for N-node episodes. Of course, in specific situations, the user may want to set a much higher threshold, but, in the absence of any special knowledge about the data, $\frac{T}{NM}$ is a meaningful initial frequency threshold to try. We illustrate this in Figs. 2 and 3 in Section 6.

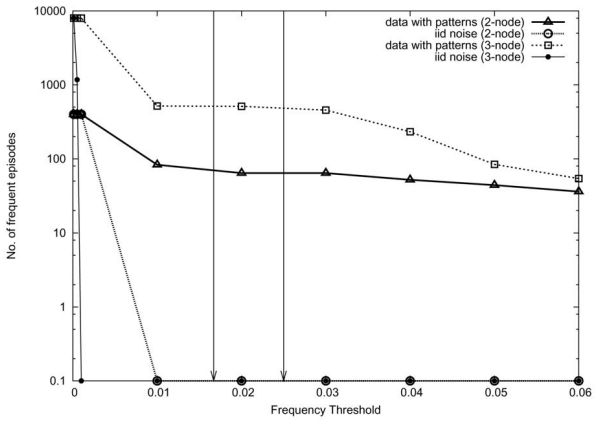

Fig. 2. The number of 2-node and 3-node frequent episodes versus the frequency threshold in synthetic data. The number of event types M = 20, data length T = 50,000. The vertical lines ending with arrows on the X-axis indicate the theoretical frequency thresholds (that is, $\frac{1}{NM}$). It is 0.025 for 2-node episodes and 0.0167 for 3-node episodes.

Thus, the method presented in this paper gives rise to what may be called *parameterless data mining*. Given only the data (in the form of an event sequence), we can discover episodes that are significant (in the sense explained earlier).

5.4 Improving Efficiency of Candidate Generation

The discussion in the previous section implies that, for the frequency measure based on nonoverlapping occurrences, there is a calculable lower bound on the frequency (which is $\frac{T}{NM}$ for N-node episodes) below which the discovered episodes are not significant. Based on the desired Type I error probability, the bound on the frequency may be higher than this, but this is certainly the minimum frequency below which the episode is not significant. An interesting aspect of this bound is that, for a smaller N, we need a larger frequency for the episode to be significant. Such a relationship is intuitively very clear and what we have here is a way of quantifying it. In analyzing large data sets, this bound can be used as a heuristic to decide, e.g., which frequent 3-node episodes need not be considered for generating candidate 4-node episodes and so on. Using such a frequency threshold, dependent on the episode length, we can improve the efficiency of candidate generation. This is illustrated empirically in Section 6.5.

6 Experimental Results

This section presents some results obtained with our new frequent episode discovery algorithm. We discuss simulation results on two types of data: synthetic data and data from some manufacturing plants.

6.1 The Data

Synthetic data was generated by embedding one or more temporal patterns or episodes (in an arbitrarily interleaved fashion) into a random stream of events. For data generation, we maintain a counter for the current time. Whenever an event is generated, it is timestamped with the current time and the counter is incremented by a small random integer. Each time, with probability ρ, the next event is generated randomly with a uniform distribution over all event types (and is termed an iid event); with the remaining

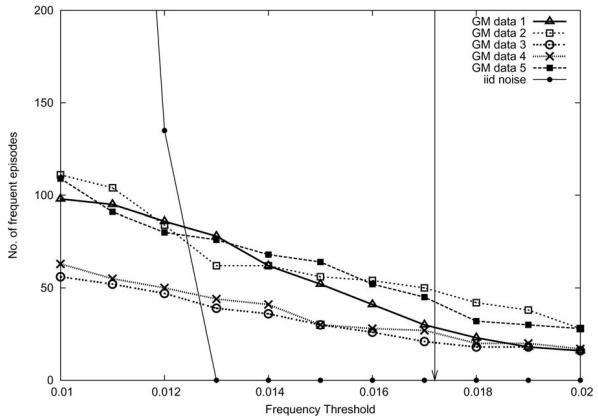

Fig. 3. The number of 2-node frequent episodes versus the frequency threshold in GM data. The number of frequent episodes for the iid noise case with frequency threshold 0.01 is 600. The vertical line ending with an arrow on the X-axis indicates the theoretical frequency threshold,

probability, it is determined by the patterns to be embedded. Whenever the next event is to be from one of the patterns to be embedded, we randomly decide between continuing with a pattern partially embedded or starting a new occurrence of a pattern. Thus, the synthetic data is like arbitrarily interleaving outputs of many EGHs and an iid noise source. Intuitively, frequent episode discovery is about finding temporally repeating patterns that may be embedded in noise. Our data contains such patterns and makes it possible to objectively assess the effectiveness of the algorithm.

The other data on which we tested our algorithm consists of time-stamped status logs of assembly lines from some manufacturing plants of General Motors (GM). For each line in each plant, the database contains (round-the-clock) information about the current status of the line recorded using an extensive set of codes. Through temporal data mining, it is possible to uncover some interesting patterns or correlations that may help in improving performance. Our new frequency count rendered fast exploration of these large data sets feasible and some interesting patterns were discovered that were regarded as useful by the data owner. Here, we provide some summary results on certain slices of the data to illustrate the efficiency and effectiveness of the algorithm.

6.2 Effectiveness of Our Frequency Count

We demonstrate the effectiveness of our algorithm by comparing the frequent episodes discovered when the event sequence was *iid* with those when we embedded some patterns. When $\rho = 1$ (that is, when events occur randomly), we expect any sequence of, say, four events to be just as frequent as any other sequence of four events. If the frequency threshold is increased starting from a low value, initially, most episodes would be frequent and, after some critical threshold, most would not be frequent. Now, suppose we embedded a few four node patterns. Then, some of their permutations and all their subepisodes would have much higher frequencies than other "random" episodes. Fig. 2 plots the number of 2-node and 3-node frequent episodes discovered by the algorithm versus the frequency threshold[2] in the two cases of *iid* noise and data

2. In all graphs, the frequency threshold is given as a fraction of data length.

TABLE 2
Rank of α and β in Sorted Frequent Episode Lists

ρ	Non-overlapping occurrences count		Windows-based count	
	α	β	α	β
0.0	1	1	1	5
0.2	1	1	1	5
0.3	1	6	1	5
0.4	1	5	1	7
0.5	1	3	1	6

with three 4-node patterns embedded in it. The sudden transitions in the graphs for the *iid* noise cases are evident. *For the iid data, the number of frequent episodes falls off to zero; however, since the plot uses log scale for the Y-axis, we showed this at 0.1 on the Y-axis.*

On the graphs, we also show, through vertical lines ending with arrows on X-axis, the theoretical frequency thresholds (cf. Section 5.3). With these thresholds, no frequent episodes are detected in the iid data, while only the embedded patterns (and their subepisodes) are detected in the other data.

Fig. 3 plots the number of 2-node frequent episodes versus the frequency threshold for five GM data sets. For the sake of comparison, the graph also shows the plot for the iid case for which a synthetic iid event sequence was generated with a similar number of event types. Notice the similarity of the plot when compared to those obtained for the synthetic data. Similar plots were obtained for the case of 3-node episodes as well. As in the case of synthetic data, in this graph also, we show the theoretical frequency threshold. (The actual value of M for different GM data sets varies between 26 and 31; the threshold shown is for $M = 29$.) Once again, the utility of our theoretical frequency threshold is evident.

6.3 Quality of Pattern Discovery

Next, we present some simulation results to show that our frequency measure is good at picking up patterns from noise. Data was generated by embedding two patterns in varying degrees of *iid* noise. The two patterns embedded were: $\alpha = (B \rightarrow C \rightarrow D \rightarrow E)$ and $\beta = (I \rightarrow J \rightarrow K)$. Data sequences with 5,000 events each were generated for different values of ρ. The respective positions of α and β (referred to as their *ranks*) in the (frequency) sorted lists of 3-node and 4-node frequent episodes discovered are shown in Table 2. We compare our algorithm with the windows-based algorithm of [1]. As can be seen from the tables, our frequency measure is as effective as the windows-based frequency in unearthing hidden temporal patterns. Also, it was observed that the sets of frequent episodes discovered by both algorithms were very similar. Similar results were observed on GM data sets also.

6.4 Efficiency of Our Algorithm

After showing that our frequency measure delivers good quality of output, we next discuss the runtimes.

TABLE 3
Runtimes (in Seconds) for Sequences of Different Lengths with Number of Event Types 20 and $\rho = 0.2$

Data length	NON-OVERLAPPING OCCURRENCE COUNT	WINDOW-BASED COUNT	Speed-up
10000	0.43	1.98	4
20000	0.58	6.45	11
30000	0.71	13.65	19
40000	0.88	25.92	29
50000	0.94	44.69	47

TABLE 4
Runtimes (in Seconds) for Different Number of Event Types with Length of Data Sequence 50,000 and $\rho = 0.2$

M	NON-OVERLAPPING OCCURRENCE COUNT	WINDOW-BASED COUNT	Speed-up
10	0.93	43.06	46
20	0.94	44.69	47
30	0.96	45.95	47
40	0.97	46.81	48
50	1.08	47.36	43

To begin with, we present runtimes[3] on synthetic data of varying lengths. Two 4-node patterns were embedded in the data generated with $\rho = 0.2$ and the total number of event types set to 20. Table 3 records runtimes of the two algorithms for five different data lengths. The frequency thresholds (which are not directly comparable for the two algorithms) were chosen so as to generate roughly the same number of frequent 4-node episodes. Not only does the nonoverlapping occurrences-based algorithm run much faster, it also scales much better with increasing sequence lengths. Table 4 gives runtimes obtained on synthetic data generated with different values for M, the number of event types. The data length in all these cases was 50,000 with two 4-node patterns embedded in them under $\rho = 0.20$. Once again, the frequency thresholds were chosen in such a way that a similar number of 4-node frequent episodes were discovered. Again, it is seen that our algorithm is faster. A similar kind of speed-up was also observed for various values of ρ with data length and number of event types kept constant. Finally, Table 5 quotes the runtimes on the GM data sets. Here also, it is noticed that our algorithm results in significant speed-up.

From the results presented here, it is clear that our frequency counting algorithm is more efficient than the windows-based counting algorithm of [1]. As was mentioned earlier, it was shown in [1] that the minimal occurrences-based count runs about 30-40 times faster than the windows-based count, but it is many times more expensive in terms of space. From the results in this section, it is clear that our algorithm is at least as fast as the minimal occurrences-based count (while it is much more efficient in terms of space).

3. All runtimes quoted are on a Pentium 4 machine with 2 GHz clock and 1 GB main memory running Linux.

TABLE 5
Runtimes (in Seconds) for GM Manufacturing Plants Data

	Data Length	NON-OVERLAPPING OCCURRENCES COUNT	WINDOWS BASED COUNT	Speed-up
Data 1	37618	0.95	158.42	166
Data 2	55191	1.41	272.67	193
Data 3	39603	1.05	157.84	150
Data 4	55959	1.64	279.36	170
Data 5	49766	1.25	177.92	142

6.5 Utility of the Episode-EGH Association

In Section 4, a connection was established between episodes and EGHs (cf. Definition 5). For establishing this connection, we need to obtain η_α, the noise parameter of the corresponding EGH. In synthetic data, if we put in only one pattern, then ρ in the data generation process would be equal to η_α. It was empirically observed that the η_α calculated was equal to ρ up to the third decimal place.

The case of synthetic data with two patterns (not sharing any event types) embedded in it is interesting. For such synthetic data (with $T = 50,000$), Table 6 lists η_α values of the most frequent episode for various values of M and ρ. Even under heavy noise (except when $\rho = 1.0$), the most frequent episode (which always turned out to be one of the patterns embedded) had η_α less than $(\frac{M}{M+1})$. Moreover, it was observed that embedded patterns (and their subepisodes) had Nf_α values greater than $\frac{T}{M}$ and the Nf_α values for all spurious patterns discovered were below $(\frac{T}{M})$. Consider the η_α values from Table 6 for $\rho = 0.4$. In this case, the data would roughly contain 40 percent noise (or iid) events and 30 percent events from each of the two episodes embedded. Since each EGH recognizing one of the episodes would treat all other events as noise, the η_α obtained is about 0.7 $(= 0.3 + 0.4)$.

In Section 5, we indicated how we can use episode-length-dependent frequency thresholds for improving the

TABLE 6
Values of η_α for Various M and ρ Values on a Synthetic Data Set with Two Episodes Embedded in It

M	$\frac{M}{M+1}$	ρ value					
		1.0	0.8	0.6	0.4	0.2	0.0
10	0.91	0.90	0.86	0.78	0.70	0.59	0.50
20	0.95	0.95	0.89	0.80	0.69	0.59	0.49
30	0.97	0.97	0.89	0.80	0.70	0.59	0.50
40	0.98	0.98	0.90	0.80	0.70	0.60	0.50
50	0.98	0.98	0.90	0.80	0.69	0.60	0.50

TABLE 7
Efficiency Gained in Candidate Generation by Using Episode Size-Dependent Frequency Thresholds

	Theoretical freq. thresh.		Freq. thresh. used		No. of freq. 3-node epsds.	No. of cand. 4-node epsds.
	3-node	4-node	3-node	4-node		
Case I	793	595	595	595	5520	62160
			793	595	2353	17808
Case II	482	361	361	361	252	707
			482	361	102	81

Case I: Synthetic data (M = 20; T = 50,000). Case II: GM data (M = 30, T = 37,618).

efficiency of candidate generation. As explained there, if the frequency of an N-node episode is less than $(\frac{T}{NM})$, then that episode can be dropped. We tested this heuristic empirically and found that it does improve efficiency. These results are summarized for one synthetic data set and one GM data set in Table 7. As can be seen from the table, by using the theoretically motivated higher threshold for 3-node episodes while keeping the same threshold for 4-node episodes, the number of candidates reduces by more than 70 percent. (In both cases, the final 4-node frequent episodes generated were the same.)

7 Conclusions

In this paper, we considered the problem of discovering frequent episodes in event streams. A new definition the for frequency of an episode was proposed, namely, the maximum number of nonoverlapping occurrences of the episode in the data stream. An algorithm for counting this frequency was presented. Through extensive simulations, it was shown that this method of frequent episode discovery is both effective and efficient. The new frequency measure yields qualitatively similar frequent episodes as the other popular method for frequent episode discovery. However, our algorithm is more efficient in terms of time taken and temporary memory needed.

A very important consequence of this new notion of frequency of an episode is that we are able to formally connect frequent episodes to a class of specialized HMMs termed EGHs. This result connects, for the first time, the frequent pattern discovery approach to statistical model learning in a formal manner. Some consequences of this have been discussed in Section 5 and we have derived an interesting test of significance for episodes based on their frequencies and have pointed out how this can lead to parameterless data mining.

Our approach for connecting frequent pattern discovery with HMM-type models can be generalized further. For example, if we look at EGHs with small values of η, then state sequences with any appreciable probability would correspond to episode occurrences that are sufficiently compact and repetitive. So, we can use η as a handle to specialize episodes to become compact or, e.g., resemble motifs which are used extensively in bioinformatics. Instead of using a delta function as the symbol probability distribution, we can use peaked distributions and then can bound the kind of "substitution error" that the system can tolerate. Often, in data constituted as event streams, different events *persist* for different durations and these time intervals may be essential for analysis. We have generalized the frequent episode framework to also handle such data [9]. In this case, we may need Semi-Markov models to obtain the necessary connections. Many of these issues will be addressed in our future research.

Acknowledgments

This research was partially funded by the GM R&D Center, Warren through SID, IISc, Bangalore. The authors thank Abhijin Adiga for help in simulations. They also thank the reviewers whose comments helped us improve the presentation.

References

[1] H. Mannila, H. Toivonen, and A.I. Verkamo, "Discovery of Frequent Episodes in Event Sequences," *Data Mining and Knowledge Discovery,* vol. 1, no. 3, pp. 259-289, 1997.
[2] R. Agrawal and R. Srikant, "Mining Sequential Patterns," *Proc. 11th Int'l Conf. Data Eng.,* Mar. 1995
[3] P. Baldi, Y. Chauvin, T. Hunkapiller, and M. McClure, "Hidden Markov Models of Biological Primary Sequence Information," *Proc. Nat'l Academy of Sciences,* vol. 91, pp. 1059-1063, Feb. 1994.
[4] *Temporal Data Mining Workshop Notes,* K.P. Unnikrishnan and R. Uthurusamy, eds., SIGKDD, Edmonton, Alberta, Canada, July 2002.
[5] C. Larizza, R. Bellazzi, and A. Riva, "Temporal Abstractions for Diabetic Patient Management," *Proc. Sixth Conf. Artificial Intelligence in Medicine in Europe,* E. Keravnou, C. Garbay, R. Baud, and J. Wyatt, eds., pp. 319-330, 1997.
[6] J. Lin, E. Keogh, S. Lonardi, and P. Patel, "Finding Motifs in Time Series," *Temporal Data Mining Workshop Notes,* K.P. Unnikrishnan and R. Uthurusamy, eds., July 2002.
[7] M. Garofalakis, R. Rastogi, and K. Shim, "Mining Sequential Patterns with Regular Expression Constraints," *IEEE Trans. Knowledge and Data Eng.,* vol. 14, pp. 530-552, May 2002.
[8] C. Bettini, X.S. Wang, S. Jajodia, and J.L. Lin, "Discovering Frequent Event Patterns with Multiple Granularities in Time Sequences," *IEEE Trans. Knowledge and Data Eng.,* vol. 10, no. 2, pp. 222-237, Mar./Apr. 1998.
[9] S. Laxman, P.S. Sastry, and K.P. Unnikrishnan, "Generalized Frequent Episodes in Event Sequences," *Temporal Data Mining Workshop Notes,* K.P. Unnikrishnan and R. Uthurusamy, eds., July 2002.
[10] J.S. Liu, A.F. Neuwald, and C.E. Lawrence, "Markovian Structures in Biological Sequence Alignments," *J. Am. Statistics Assoc.,* vol. 94, pp. 1-15, 1999.
[11] D. Chudova and P. Smyth, "Pattern Discovery in Sequences under a Markovian Assumption," *Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* July 2002.
[12] D.L. Wang and B. Yuwono, "Anticipation-Based Temporal Pattern Generation," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 25, no. 4, pp. 615-628, 1995.
[13] F. Korkmazskiy, B.H. Juang, and F. Soong, "Generalized Mixture of HMMs for Continuous Speech Recognition," *Proc. 1997 IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP-97),* vol. 2, pp. 1443-1446, Apr. 1997.
[14] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic, "Discovering Clusters in Motion Time Series Data," *Proc. 2003 IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. I-375-I-381, June 2003.
[15] P. Smyth, "Data Mining at the Interface of Computer Science and Statistics," *Data Mining for Scientific and Engineering Applications,* R.L. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar, and R.R. Namburu, eds. Kluwer Academic Publishers, 2001.

[16] G. Casas-Garriga, "Discovering Unbounded Episodes in Sequential Data," *Proc. Seventh European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD '03),* pp. 83-94 2003.
[17] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE,* vol. 77, pp. 257-286, Feb. 1989.
[18] B. Gold and N. Morgan, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music.* New York: John Wiley & Sons, Inc., 2000.
[19] R. Gwadera, M.J. Atallah, and W. Szpankowski, "Reliable Detection of Episodes in Event Sequences," *Proc. Third IEEE Int'l Conf. Data Mining (ICDM '03),* pp. 67-74 Nov. 2003.
[20] M.J. Atallah, R. Gwadera, and W. Szpankowski, "Detection of Significant Sets of Episodes in Event Sequences," *Proc. Fourth IEEE Int'l Conf. Data Mining (ICDM '04),* pp. 3-10, Nov. 2004.
[21] R. Gwadera, M.J. Atallah, and W. Szpankowski, "Markov Models for Identification of Significant Episodes," *Proc. 2005 SIAM Int'l Conf. Data Mining (SDM-05),* Apr. 2005.
[22] P. Flajolet, Y. Guivarc'h, W. Szpankowski, and B. Vallee, "Hidden Pattern Statistics," *Proc. 28th Int'l Colloquium Automata, Languages, and Programming,* pp. 152-165, 2001.

Srivatsan Laxman received BE degree in electronics and communications engineering from the University of Mysore in 1998 and the MS degree in system science and signal processing from the Indian Institute of Science, Bangalore, in 2002. He is currently a PhD student in the Department of Electrical Engineering, Indian Institute of Science, Bangalore. His research interests include machine learning, pattern recognition, data mining, and signal processing.

P.S. Sastry (S'82-M'85-SM'97) received the PhD degree in electrical engineering from the Indian Institute of Science, Bangalore, in 1985. Since 1986, he has been on the faculty of the Department of Electrical Engineering, Indian Institute of Science, Bangalore, where he is currently a professor. He has held visiting positions at the University of Massachusetts, Amherst, the University of Michigan, Ann Arbor, and General Motors Research Labs, Warren, Michigan. His research interests include learning systems, pattern recognition, data mining, and image processing. He is a senior member of the IEEE.

K.P. Unnikrishnan received the PhD degree in physics (biophysics) from Syracuse University, Syracuse, New York, in 1987. He is currently a staff research scientist at the General Motors R&D Center, Warren, Michigan. Before joining GM, he was a postdoctoral member of the technical staff at AT&T Bell Laboratories, Murray Hill, New Jersey. He has also been an adjunct assistant professor at the University of Michigan, Ann Arbor, a visiting associate at the California Institute of Technology (Caltech), Pasadena, and a visiting scientist at the Indian Institute of Science, Bangalore. His research interests concern neural computation in sensory systems, correlation-based algorithms for learning and adaptation, dynamical neural networks, and temporal data mining.